

关注微信公众号: PMvideo

# 【Spring学习04】实例化上下文对象及加载多个配置文件

作者: soonfly (/authorarticle.html?author=soonfly) 2017-03-30 ☆ 收录到我的专题 (/select.html?articleId=401750)

标签 配置文件 (<http://www.bijishequ.com/info/search.html?searchText=配置文件>) 加载 (<http://www.bijishequ.com/info/search.html?searchText=加载>) newClassPathXmlApplicationContext (<http://www.bijishequ.com/info/search.html?searchText=newClassPathXmlApplicationContext>) applicationContextxml (<http://www.bijishequ.com/info/search.html?searchText=applicationContextxml>) http (<http://www.bijishequ.com/info/search.html?searchText=http>)

本文博客地址: <http://blog.csdn.net/soonfly/article/details/68500062> (<http://blog.csdn.net/soonfly/article/details/68500062>) (转载请注明出处)

## 一、实例化上下文对象

从前面的例子可以看到, 在代码中实例化上下文对象非常简单, 如下面的例子:

```
1 ApplicationContext ctx = new ClassPathXmlApplicationContext( "spring-beans.xml");
2 FooService foo = (FooService) ctx.getBean("FooService");
```

几个常用的类:  
**ClassPathXmlApplicationContext:**从类路径下的xml配置文件中加载上下文定义.  
**FileSystemXmlApplicationContext:**读取文件系统下xml配置文件并加载  
**XmlWebApplicationContext:**读取Web应用下的Xml配置文件并加载上下文定义

ClassPathXmlApplicationContext与FileSystemXmlApplicationContext两者的区别只是查找配置文件的起始路径不同, 一个以classpath为当前路径, 一个是直接用文件系统的当前路径. 内部没有太大区别。  
XmlWebApplicationContext同ClassPathXmlApplicationContext一样, 只不过应用于WEB工程项目。

## 二、加载多个配置文件

开发中, 我们经常以业务功能或业务分层的方式, 定义不同的xml配置文件。  
如何加载多个xml配置文件呢? 可以这样:

```
1 ApplicationContext context = new ClassPathXmlApplicationContext(new String[] { "Spring-Common.xml", "Spring-Connection.xml", "Spring-ModuleA.xml" });
```

但是这种方法不易组织并且不好维护。

如果配置文件存在多个的情况下, 推荐的加载配置文件的方式有以下几个:

1、指定总的配置文件去包含子的配置文件, 然后只加载总的配置文件即可。  
在总的配置文件applicationContext.xml中使用import标签进行子文件包  
总配置文件applicationContext.xml:

```
1 <beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
2 schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-
3 3.0.xsd">
4
5 <import resource="common/Spring-Common.xml"/>
6 <import resource="connection/Spring-Connection.xml"/>
  <import resource="moduleA/Spring-ModuleA.xml"/>
  </beans>
```

代码中加载配置文件:

```
1 ApplicationContext ac= new ClassPathXmlApplicationContext("applicationContext.xml");
```

2、使用星号来匹配多个文件进行加载,文件名称要符合规律。(推荐使用)

配置文件 的名称如下：  
applicationContext.xml  
applicationContext-action.xml  
applicationContext-service.xml  
applicationContext-dao.xml

```
1 ApplicationContext ac = new ClassPathXmlApplicationContext("applicationContext*.xml");
```

3、可以使用数组作为参数，一次性加载多个配置文件

```
1 String[] files={"applicationContext.xml","applicationContext-test.xml"};  
2 ApplicationContext ac = new ClassPathXmlApplicationContext(files);
```

注意：自动装配功能和手动装配要是同时使用，那么自动装配就不起作用。

4、JavaConfig配置的情况下：可单独创建一个AppConfig.java，然后将其他的配置导入到AppConfig.java中

```
1 import org.springframework.context.annotation.Configuration;  
2 import org.springframework.context.annotation.Import;  
3  
4 @Configuration  
5 @Import({ CustomerConfig.class, SchedulerConfig.class })  
6 public class AppConfig {  
7  
8 }
```

这样，加载时，只需要加载AppConfig.java即可

```
1 ApplicationContext context =  
2     new AnnotationConfigApplicationContext(AppConfig.class);
```

本文博客地址： <http://blog.csdn.net/soonfly/article/details/68500062> (<http://blog.csdn.net/soonfly/article/details/68500062>)（转载请注明出处）



登录

(/logi

n.htm

l?redi

rectU

rl=%

2Fdet

91862

F401

750)

后发表评论

0条评论或问题

社区邀请

笔记社区是一个面向中高端IT开发者、程序员的知识共享社区，通过网络抓取与文章分类总结，由专家为用户提供高质量的专题文章系列。

邀请您成为社区专家 >> (<http://www.bijishequ.com/creat.html>)

原文链接：<http://blog.csdn.net/soonfly/article/details/68500062>

声明：所有文章资源均从网络抓取，如果侵犯到您的著作权，请联系删除文章。联系方式请关注微信公众号PMvideo【锤子视频-程序员喜欢的短视频】，或者加笔记社区开发者交流群 628286713。

## 4 签到

January 今日签到1人

相关标签

- Struts (/tag/list?tagId=1440)
- loc (/tag/list?tagId=1441)
- Spring (/tag/list?tagId=1438)
- SpringMVC (/tag/list?tagId=1560)
- Web服务 (/tag/list?tagId=1619)
- ATOM (/tag/list?tagId=1623)
- Xml (/tag/list?tagId=1621)
- IWorkbench (/tag/list?tagId=519)
- EnumMap (/tag/list?tagId=518)
- HttpRequest (/tag/list?tagId=516)
- StringUtils (/tag/list?tagId=515)
- RSS (/tag/list?tagId=1622)
- SOAP (/tag/list?tagId=1620)
- BufferedInputStream (/tag/list?tagId=72)
- Configuration (/tag/list?tagId=429)
- PriorityQueue (/tag/list?tagId=772)
- IEditorPart (/tag/list?tagId=773)
- FilterInputStream (/tag/list?tagId=2469)
- After (/tag/list?tagId=69)
- TrustManager (/tag/list?tagId=427)
- Context (/tag/list?tagId=771)
- Preconditions (/tag/list?tagId=426)
- NoSuchAlgorithmException (/tag/list?tagId=73)
- JavaArchive (/tag/list?tagId=770)
- HttpServletResponse (/tag/list?tagId=70)
- Shader (/tag/list?tagId=769)
- AbstractModule (/tag/list?tagId=509)
- JavaPlugin (/tag/list?tagId=703)
- MenuItem (/tag/list?tagId=704)
- IStructuredContentProvider (/tag/list?tagId=975)
- ImmutableList (/tag/list?tagId=512)
- SensorManager (/tag/list?tagId=974)